(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 April 2001 (05.04.2001)

PCT

(10) International Publication Number
WO 01/24032 A2

(51) International Patent Classification^7: G06F 17/00

(21) International Application Number: PCT/US00/19731

(22) International Filing Date: 20 July 2000 (20.07.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/404,597 24 September 1999 (24.09.1999) US

(71) Applicant: NET PERCEPTIONS, INC. [US/US]; 7901 Flying Cloud Drive, Eden Prairie, MN 55344 (US).

(72) Inventors: RAUSER, John; 2720 Aldrich Avenue S, Minneapolis, MN 55408 (US). GURALNIK, Valerie; 7235 Stewart Drive, Eden Prairie, MN 55346 (US).

(74) Agents: GARRETT, Arthur, S. et al.; Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., 1300 I Street, N.W., Washington, DC 20005-3315 (US).

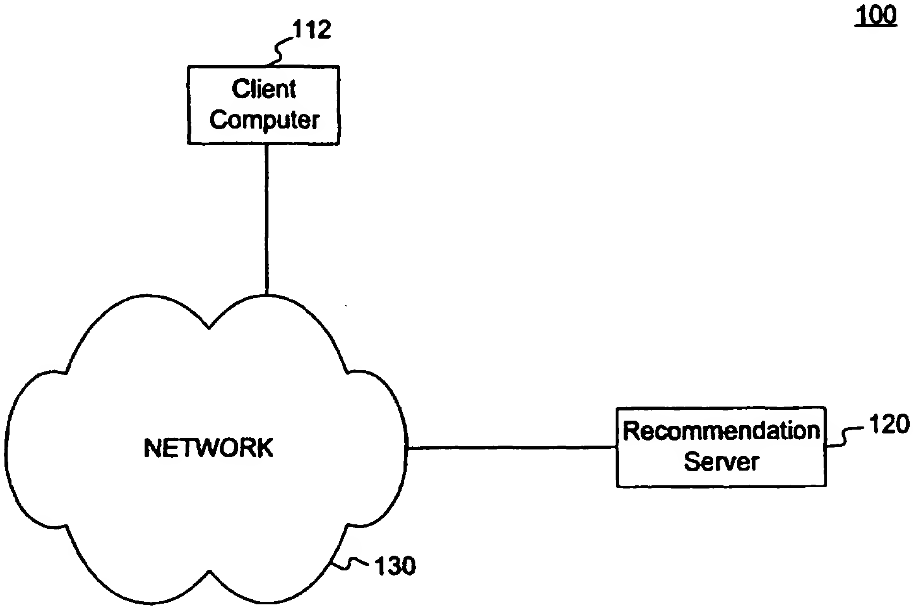
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: OPTIMIZED RULE BASED CONSTRAINTS FOR COLLABORATIVE FILTERING SYSTEMS



(57) Abstract: Methods, systems, and articles of manufacture consistent with the present invention provide a recommendation server that receives a recommendation request from a user of a client computer. The recommendation server contains software to provide recommendations to the user. To provide the recommendations, the recommendation server applies a constraint filter and a recommendation filter on a set of items.

WO 01/24032 A2

PAT-009B
74

OPTIMIZED RULE BASED CONSTRAINTS FOR COLLABORATIVE FILTERING SYSTEMS

Background of the Invention

Field of the Invention

This invention relates generally to data processing systems, and more particularly, collaborative filtering and recommender systems.

Description of the Related Art

Recommender systems are becoming widely used in e-commerce business activities. For example, systems that make personalized recommendations are used as a marketing tool to turn "window shoppers" into buyers, increase cross-sells and up-sells, and deepen customer loyalty. Recommender systems allow e-commerce operators to take advantage of customer databases to provide valuable personalized service to customers.

Current recommender systems can make generic recommendations to customers, but they do not take into account many of the business rules that merchandisers wish to implement, such as "don't recommend an item that is out of stock," "don't recommend an item from a category that the customer has not selected,""don't recommend items that are not in season," or "don't recommend inappropriate items to minors." In other words, current recommender systems base recommendations solely on the customer preference data.

Existing recommender systems allow only the simplest form of filtering, and they do it one of two ways, prefiltering or postfiltering.

Prefiltering requires a constraint system that discovers acceptable items and then submits all discovered items to a prediction system that makes recommendations from this subset. Prefiltering has some serious practical limitations, however. For example, gathering the list of acceptable items is difficult to accomplish efficiently as the list of acceptable items may be very large since it is selected from the whole item catalog.

Postfiltering also requires a system to filter the recommendation list. Postfiltering requires that the recommendation system produce more recommendations than actually required. The oversized list is passed to a constraint system, which then removes unacceptable items. Although postfiltering may avoid the problem of having to select items from a large list, it may fail to provide recommendations if the postfiltering eliminates all items.

Summary of the Invention

Methods and systems consistent with the present invention provide a recommendation server that receives a recommendation request from a user of a client computer. The recommendation server contains software to provide recommendations to the user. To provide the recommendations, the recommendation server applies a constraint filter and a recommendation filter to a set of items.

In accordance with methods and systems consistent with the present invention, a method for providing a recommendation list specifies a constraint filter to select items satisfying a constraint, selects the items that satisfy the constraint filter, computes predicted values based on a recommendation filter, and appends the items meeting predetermined criteria.

In accordance with methods and systems consistent with the present invention, a method for applying a recommendation filter and a constraint filter to a plurality of items is provided. The method receives a recommendation request from a user, specifies a constraint filter to select ones of the items satisfying a constraint, and determines the order of the filters based on a cost of the filters. The method applies the constraint filter first when the cost of the constraint filter is lower than the cost of the recommendation filter. Otherwise, the method applies the recommendation filter first when the cost of the recommendation filter is lower than the cost of the constraint filter.

Brief Description of the Drawings

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an implementation of the invention and, together with the description, serve to explain the advantages and principles of the invention. In the drawings,

Figure 1 depicts a data processing system suitable for practicing methods and systems consistent with the present invention;

Figure 2 depicts a more detailed diagram of the client computer depicted in Fig. 1;

Figure 3 depicts a more detailed diagram of the recommender server depicted in Fig. 1;

Figure 4 depicts a flow chart of the steps performed by the data processing system of Fig. 1 when initiating the constraint process consistent with methods and systems of the present invention;

Figure 5 depicts a flow chart of the steps performed by the data processing system of Fig. 1 when initiating the recommender process in accordance with methods and systems consistent with the present invention;

Figure 6A depicts a constraint tree consistent with methods and systems of the present invention;

Figure 6B depicts an recommendation request form interface consistent with methods and systems of the present invention;

Figure 6C depicts an output recommendation list interface consistent with methods and systems of the present invention; and

Figure 7 depicts a constraint filter and recommendation filter consistent with methods and systems of the present invention.

Detailed Description

The following detailed description of the invention refers to the accompanying drawings. Although the description includes exemplary implementations, other implementations are possible, and changes may be made to the implementations described without departing from the spirit and scope of the invention. The following detailed description does not limit the invention. Instead, the scope of the invention is defined by the appended claims. Wherever possible, the same reference numbers will be used throughout the drawings and the following description to refer to the same or like parts.

Overview

Recommender systems provide recommendations to users based on various attributes. For example, collaborativc filtering (CF) systems are a specific type of recommender system that recommend items to a user based on the opinions of other users. In their purest form, CF systems do not consider the content of the items at all, relying exclusively on the judgement of humans of the item's value. In this way, CF systems attempt to recapture the cross-topic recommendations that are common in communities of people.

Commercial applications of ratings-based collaborative filtering now exist in a variety of domains including books, music, grocery products, dry goods, and information. One example of a CF system is the GroupLens Research system that provides a CF for Usenet news and movies. More information on CF technology may be found at http://www.netperceptions.com, hereby incorporated by reference.

To use the recommendation system, an operator may first create a constraint using a constraint language that allows different business rules to be described in textual form. For example, to select a candidate from a set of red items, a constraint may be: "candidate isa red-thing." To select a candidate from a set of movies that are both comedies and not r-rated, a constraint may be: "candidate isa comedy and not candidate isa r-rated."

An item may be anything for which a user may recommend. For example, in the domain of movies, each movie may be an item. An itcm may be assigned arbitrarily to one or more categories. For example, a fiction book may be a member of the "Fiction" category. Category membership may represent any attribute of a user or item. For example, an item that is in stock may be a member of the "in stock" category or an itcm that is red may be a member of the "red-things" category. This type of categorization allows the recommendation system to apply a constraint filter based on any attribute or combination of attributes of the item. A constraint filter is a software with a complex boolean expression as an attribute that the recommendation system uses to restrict items.

A constraint may also consist of free variables. A free variable is a placeholder for an attribute that can be determined at execution time. For example, to provide the user with the ability to choose a category when applying a constraint, a constraint may be: "candidate isa X," where the user inputs X at runtime.

Once the operator creates the constraint, the recommendation system may begin accepting recommendation requests from a user. To use the recommendation system, a user may access a web site with instructions and web pages for the user to fill out to use the recommendation system. Upon accessing the web site, the user enters a recommendation request including values for various free variables and the number of items desired. The recommendation request may include the values for the free variables for the constraint filter, the number of items desired and user information. One skilled in the art will appreciate that other methods may be used to obtain a recommendation request, such as a telephone call center or manual entry.

Once the recommendation system receives the recommendation request, the system next determines the order of a constraint filter and a recommendation filter to apply to the received recommendation request. The constraint filter determines whether an item satisfies a constraint and the recommendation filter determines a predicted value of the item based on historical or statistical information. One type of recommendation filter is the well-known collaborative filtering (CF) technique.

The recommendation filter may compute a predicted value to determine if an item should be recommended. A predicted value is a number that rates an item according to certain criteria. For example, a predicted value may be used to rank an item based on recommendations from similar users of the recommendation system. The predicted value is essentially an estimate of how much a user is likely to enjoy an item and may be determined, for example, by a CF technique. One skilled in the art will appreciate that the predicted value may be determined in a number of different ways, such as previous purchases, previous comments or a particular rating given by the user.

To determine the order of filters to apply, the recommendation system determines the cost of applying successively each filter to all items. The cost of applying each filter is explained below. For example, if the cost to apply a constraint filter before a recommendation filter is lower than the other sequence, the recommendation system may choose to apply the constraint filter first. Preferably, the recommendation system provides a recommendation to a user with the lowest cost.

Once the order of the filters is determined, the recommendation system recommends a list of items to the user that the user may be interested in based on the recommendation filter and also passes the constraint filter. If the cheapest method is to apply the constraint filter first, each time an item passes the constraint, it becomes a potential candidate for a recommendation list. The candidate is then passed to the recommendation filter. If the candidate passes the recommendation filter, the candidate and the candidate's predicted value are appended to a recommendation list.

System Components

Fig. 1 depicts a data processing system 100 suitable for practicing methods and systems consistent with the present invention. Data processing system 100 comprises a client computer 112 connected to recommendation server 120 via a network 130, such as the Internet. The user uses client computer 112 to provide various information to recommendation server 120.

Recommendation server 120 transmits and receives web pages from a browser on client computer 112 using hypertext markup language (HTML), Java or other techniques. These web pages may include images or instructions to obtain recommendation requests from a user. Recommendation server 120 also contains a database that stores various data, such as constraint filters, recommendation filters and items, further described below.

Although only one client computer 112 is depicted, one skilled in the art will appreciate that data processing system 100 may contain many more client computers and additional client sites. One skilled in thc art will also appreciate that client computer 112 may come with the recommendation server software already installed.

Figure 2 depicts a more detailed diagram of client computer 112, which contains a memory 220, a secondary storage device 230, a central processing unit (CPU) 240, an input device 250, and a video display 260. Memory 220 includes browser 222 that allows users to interact with recommendation server 120 by transmitting and receiving files. An example of a browser suitable for use with methods and systems consistent with the present invention is the Netscape Navigator browser, from Netscape.

As shown in Figure 3, recommendation server 120 includes a memory 310, a secondary storage device 320, a CPU 330, an input device 340, and a video display 350. Memory 310 includes recommendation software 312, which determines if an item should be recommended to the user by applying a constraint filter 316 and a recommendation filter 318, described below. Recommendation software 312 also interacts with client computer 112 for transmitting and receiving files, such as HTML files or Java files. To interact with client computer 112, recommendation software may include a web server. Although a web server is described in this particular embodiment of the recommendation server, recommendation server 120 may interact with a client in other ways such as, voice prompts, call centers, or kiosks. Memory 310 also includes constraint builder software 314, which creates constraints that are used by recommendation software 312 to recommend an item to the user. Recommendation software 312 and constraint builder software 314 also provide access to database 322 in secondary storage device 320.

Secondary storage device 320 includes grammar file 322 containing a set of rules which map textual constraints to their internal representation in the constraint filter. Secondary storage device also includes database 324 with constraint table 326 that stores built constraints to use with recommendation software 312 and item table 328 with attribute information about each item. For example, item table 328 could store a category identification, item number, and number in stock.

Although aspects of the present invention are described as being stored in memory, one skilled in the art will appreciate that these aspects may be stored on or read from other computer-readable media, such as secondary storage devices, like hard disks, floppy disks, and CD-ROM; a carrier wave received from a network like the Internet; or other forms of ROM or RAM. Additionally, although specific components and programs of client computer 112 and recommendation server 120 have been described, one skilled in the art will appreciate that these may contain additional or different components or programs.

Constraint Creation Process

Figure 4 depicts a flow chart of the steps performed when creating a constraint on recommendation server 120. The constraint creation process is initiated, for example, by an operator inputting a textual constraint into constraint builder software 314 (step 402). The constraint may contain free variables or bound expressions. After the operator inputs the constraint, the builder software checks the syntax of the constraint (step 404). For example, an acceptable syntax may include logical expressions or relational expression. That is, constraint builder software verifies that the operator created a valid constraint to signify a possible business rule. Logical expression include, for example, AND, OR, or NOT boolean expressions. Relational expressions include, for example EQUAL TO, GREATER THAN, LESS THAN or ISA.

Once the constraint is verified for the correct syntax, constraint builder software may translate the textual constraint into, for example, a constraint in a tree structure format (step 406). Constraint builder software 314 includes the well-known yacc parser to translate the textual constraint. The builder software reads grammar specification file 322 and generates a constraint tree consisting of objects by using the grammar specifications applied to the textual constraint. The new constraint tree is in a format acceptable to recommendation software 312.

Grammar specification file 322 consists of many different object formats to create the constraint tree, such as logical expression objects, relational expression objects or leaf objects. Logical expression objects are a type of boolean expression, such as AND, OR or NOT. Relational expression objects compare two leaf objects. Leaf objects represent any entity in the application domain. A leaf may be a category leaf, free variable leaf, candidate leaf, or subject leaf. A category leaf represents at least one item in database 324, such as"Thriller" or "Shoes." A free variable leaf is essentially a placeholder that is specified at execution time by the user or operator. The free variable leaf stores a variable name, which is bound to an actual entity in the application domain during the recommendation process, further described below. Similar to the free variable leaf, the candidate leaf is also a placeholder. The candidate leaf represents the actual item discovered by the recommendation filter. Finally, a subject leaf is a placeholder for the user who requested the recommendation. For example, a user may have an attribute of being 14 years old. Thus, leaf objects in the constraint tree may reflect this attribute.

For example, a textual constraint created by an operator to produce recommendations for movies that are now playing, and are of a genre selected by a user, and prohibits the recommendation of r-rated movies to minors may be:

X: (candidate isa movie) and (candidate isa X) and (candidate isa NowPlay) and (not (subject isa minor) and (candidate isa R-rated)).

Constraint builder software 314 would translate the textual constraint to the tree of objects depicted in Figure 6A.

Once constraint builder software 314 translates the textual constraint to a constraint tree, the constraint tree is placed as an entity in constraint table 326 (step 408). The constraint tree is used later by recommendation software 312 during the recommendation process. This completes the constraint creation process.

Recommendation Process

Figure 5 depicts a flow chart of the steps performed when initiating the recommendation process in accordance with methods and systems consistent with the present invention. The recommendation process is initiated, for example, by a user accessing recommendation server 120 (step 502). Once accessed, recommendation software 312 transmits a recommendation request page to client computer 112 (step 504). The request page may be in HTML. One skilled in the art will appreciate that the inquiry page may be designed in other formats, such as Visual Basic or Java. The request page may include a category selection fields 602 and 604, desired results field 606 for the user to fill out, and a submit button 608, as shown in Figure 6B. Once the request page is displayed on browser 222, the user may select a category and enter the number of results and submit the request to recommendation server 120 by pressing button 608 (step 506).

When button 608 is pressed, browser 222 transmits the category and number of results information to recommendation server 120 using the well-known HyperText Transport Protocol (HTTP).

Once received at recommendation server 120, recommendation software 312 binds the free variables in the appropriate constraint with the category selected by the user and the number of items desired (step 508). To do so, recommendation software 312 first locates the appropriate constraint in constraint table 326. The constraint may be found in constraint table 326, for example, by a tag appended to the recommendation request indicating the constraint. Once located, recommendation software 312 descends the constraint tree to locate free variable objects. Once a free variable object is found, recommendation software 312 copies the information from the recommendation request to the free variable. For example, in Fig. 6B, if a user selected "rock" and "jazz" as the categories to search, step 508 generates an array of length two that contains the object representing jazz and rock with a corresponding index number. The array looks as follows:

Index	Object
1	Jazz
2	Rock

Step 508 uses the array when descending the constraint tree searching for a free variable object. When a free variable object is located, the index is matched and thc corresponding object is copied to the free variable object. Ultimately, recommendation software 312 will examine each node in the constraint tree to locate all free variables and store corresponding information in each free variable. Although two free variables were used in Fig. 6B, one skilled in the art will appreciate that many more free variables may exist in the recommendation request.

After thc free variables in the constraint tree are bound, recommendation softwarc 312 examines each item in item table 328 for an item to recommend to the user. The process begins with recommendation software determining the lowest cost method to complete a recommendation request (step 510). To do so, recommendation software 312 determines the cost of applying constraint filter 316 and recommendation filter 318 in different orders to the items. As shown in Fig. 7, each filter has a generation interface that produces items and a rejection interface that determines whether a particular item is suitable for recommendation to the user. Each filter is applied sequentially. The generation interface is called on the first filter, and the produced items are passed to the rejection interface on the second filter.

If the cost of generating a sufficient number of items by applying the generation interface of the constraint filter before applying the rejection interface of the recommendation filter is lower than applying the generation interface of the recommendation filter before applying the rejection interface of the constraint filter, then recommendation software 312 will apply the generation interface of the constraint system first to item table 328. Otherwise, recommendation software 312 applies the generation interface of the recommendation filter first. The cost may be approximated by the following equation:

> Cost = (number of results required / probability that a randomly selected item will pass the rejection interface of the second filter) * (cost of applying the generation interface of the first filter to generate a single item + cost of applying the rejection interface of the second filter to a single item)

Once the order of the filters is determined, recommendation software 312 determines if enough items have been located (step 512). That is, recommendation software continues to discover new items in item table 328 until the required number of items requested from the user has been reached. Once an item has been discovered in item table 328, the item is evaluated (step 514). Evaluation occurs by applying the constraint filter to the item. Items that pass the constraint filter will be passed to the recommendation filter (step 516). An item passes the constraint filter when it satisfies the constraints conditions. If an item does not pass the constraint filter, the item is discarded and not recommended.

Next, the recommendation filter may compute a predicted value for the item (step 518). Also in step 518, each item whose predicted value is at least a threshold value is appended to a result list for display on client computer 112. The results may be displayed in HTML.

Figure 6C depicts an output interface 620 presented to the user after submitting the recommendation request in Fig. 6B. Output interface 620 contains a recommendation list 622. For example, the user may select an item from the list to purchase.

Conclusion

Methods, systems, and articles of manufacture consistent with the present invention provide a recommendation server that rcccives a recommendation request from a user of a client computer. The recommendation server contains software to provide recommendations to the user. To provide the recommendations, the recommendation server applies a constraint filter and a recommendation filter on a set of items.

The foregoing description of an implementation of the invention has been presented for purposes of illustration and description. It is not exhaustive and does not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but the present invention may be implemented as a combination of hardware and software or in hardware alone.

CLAIMS

1. A method for providing a recommendation list from a plurality of items, comprising the steps of executed in a data processing system:

specifying a constraint filter to select ones of the items satisfying a constraint;

selecting the ones of the plurality of items that satisfy the constraint filter;

computing predicted values based on a recommendation filter, for the selected ones of the items; and

appending the selected ones of the items meeting predetermined criteria.

2. The method of claim 1, wherein appending selected ones of the items further includes appending the selected ones of the items to the recommendation list when the predicted value exceeds a predetermined number.

3. The method of claim 1, wherein appending selected ones of the items further includes appending a predetermined number of items to the list.

4. The method of claim 1, wherein selecting the ones of the items that are restricted further includes applying a constraint containing free variables to the ones of the items.

5. The method of claim 1, wherein selecting the ones of the items that are restricted further includes applying a constraint containing bound expressions.

6. The method of claim 1, wherein selecting the ones of the items that are restricted further includes applying a boolean constraint filter.

7. The method of claim 1, wherein selecting the ones of the items that are restricted further includes applying a constraint to the ones of the items item, wherein the constraint signifies an equality.

8. The method of claim 1, wherein selecting the ones of the items that restricted further includes applying a constraint to the ones of the items, wherein the constraint signifies category membership.

9. The method of claim 1, wherein computing a predicted value further includes evaluating the selected ones of the items with collaborative filtering.

10. The method of claim 1, further comprising the step of:

truncating the recommendation list when a predetermined number of the selected ones of the items on the recommendation list has been met.

11. The method of claim 1, wherein selecting the ones of the items that are restricted by the constraint filter further includes

obtaining data from a user; and

adding the data to the constraint filter.

12. The method of claim 1, wherein specifying a constraint filter further includes

obtaining the constraint from an operator; and

storing the constraint filter in memory.

13. A method for applying a recommendation filter and a constraint filter to a plurality of items in a data processing system, comprising the steps of:

receiving a recommendation request from a user;

specifying a constraint filter to select ones of the items satisfying a constraint;

determining the order of the filters based on a cost of the filters;

applying the constraint filter first when the cost of the constraint filter is lower than the cost of the recommendation filter; and

applying the recommendation filter first when the cost of the recommendation filter is lower than the cost of the constraint filter.

14. The method of claim 13, wherein determining the order of the filters further includes

analyzing the cost for applying the filters to the ones of the items; and

determining the probability that the ones of the items will pass both filters.

15. The method of claim 14, further comprising:

solving the following equation for the cost to apply to each filter:

$C = M / P (G + R)$,

wherein M is a number of desired items, P is the probability that the ones of the items will pass the second filter, G is the time to retrieve the ones of the items, and R is the time to decide if the ones of the items will pass the second filter.

16. An apparatus designed to provide a recommendation list from a plurality of items in a data processing system, comprising:

a processing component configured to process instructions for:

applying a constraint filter to ones of the items;

applying a recommendation filter to ones of the items, and

determining an order of the filters to apply to the plurality of the items; and

a recommender component configured to append the filtered ones of the items to a recommendation list based on the constraint filter and recommendation filter.

17. The apparatus of claim 16, wherein the processing component computes predicted values based on the recommendation filter.

18. The apparatus of claim 16, wherein the processing component further

determines the order of the filters to apply to the plurality of the items based on the cost of the filters;

applies the constraint filter first when it is determined that the cost of the constraint filter is lower than the cost of the recommendation filter; and

applies the recommendation filter first when it is determined that the cost of the recommendation filter is lower than the cost of the constraint filter.

19. The apparatus of claim 16, wherein the processing component applies a constraint filter based on a boolean constraint containing free variables.

20. The apparatus of claim 16, wherein the processing component applies a constraint filter containing bound expressions.

21. The apparatus of claim 16, wherein the processing component applies a constraint filter based on a boolean constraint.

22. The apparatus of claim 16, wherein thc processing component applies a constraint filter that signifies category membership.

23. The apparatus of claim 16, wherein the processing component applies a constraint filter that signifies an equality.

24. The apparatus of claim 16, wherein the processing component computes predicted values by evaluating each ones of the items with collaborative filtering.

25. The apparatus of claim 16, wherein the recommender component is further configured to truncate the recommendation list when a predetermined number of the ones of the items on the recommendation list has been met.

26. The apparatus of claim 16, further comprising an input component configured to:
obtain the constraint from an operator; and
store the constraint filter in a memory.

27. The apparatus of claim 16, further comprising an input componcnt configured to:
obtain data from a user; and
add the data to the constraint filter.

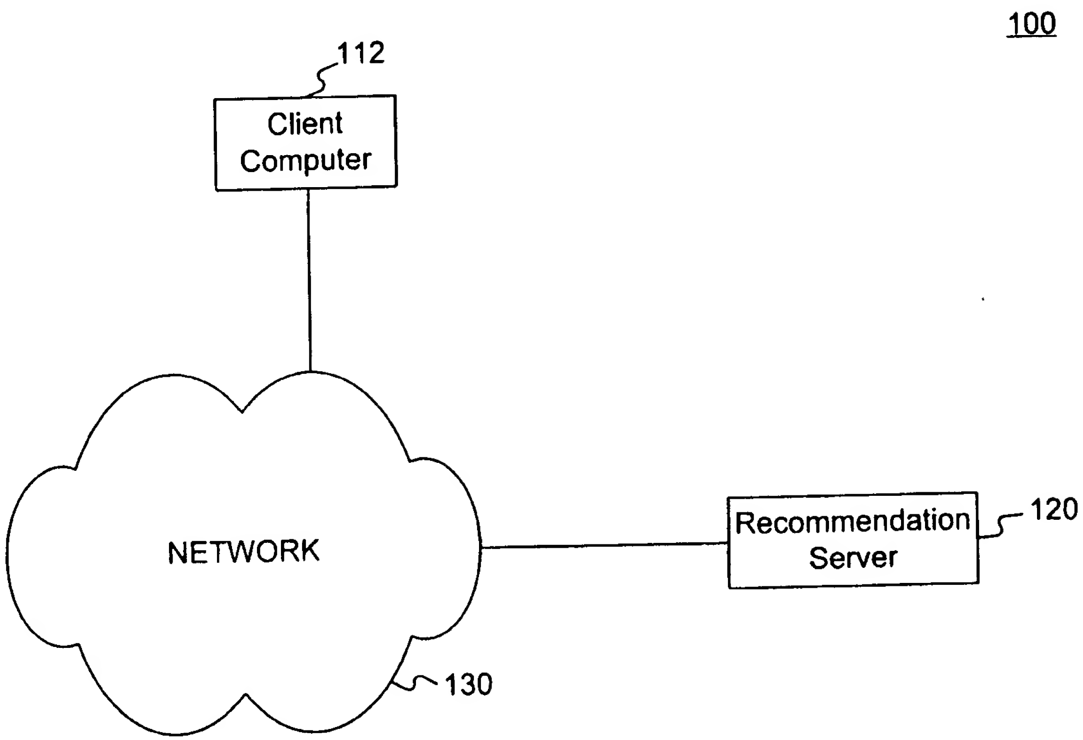

100
112
Client Computer
NETWORK
130
Recommendation Server
120

FIG. 1

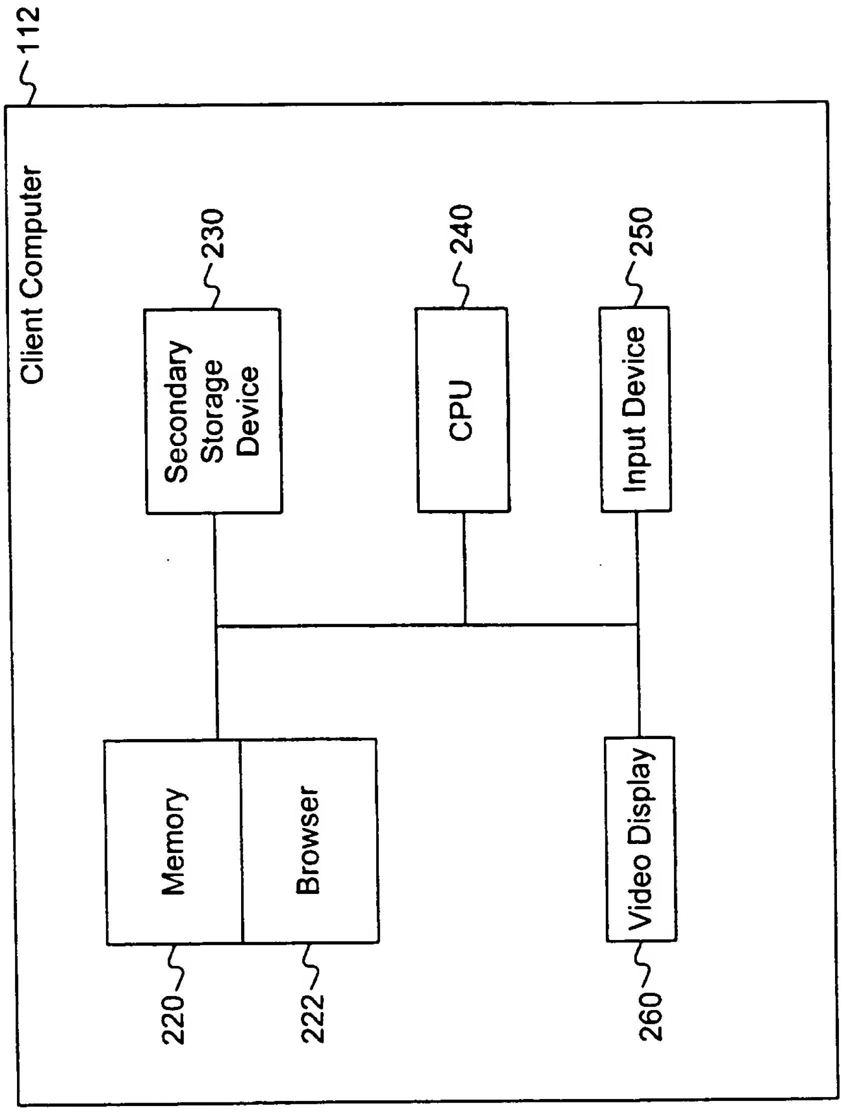

112
Client Computer
Secondary Storage Device
230
CPU
240
Input Device
250
Memory
220
Browser
222
Video Display
260

FIG. 2

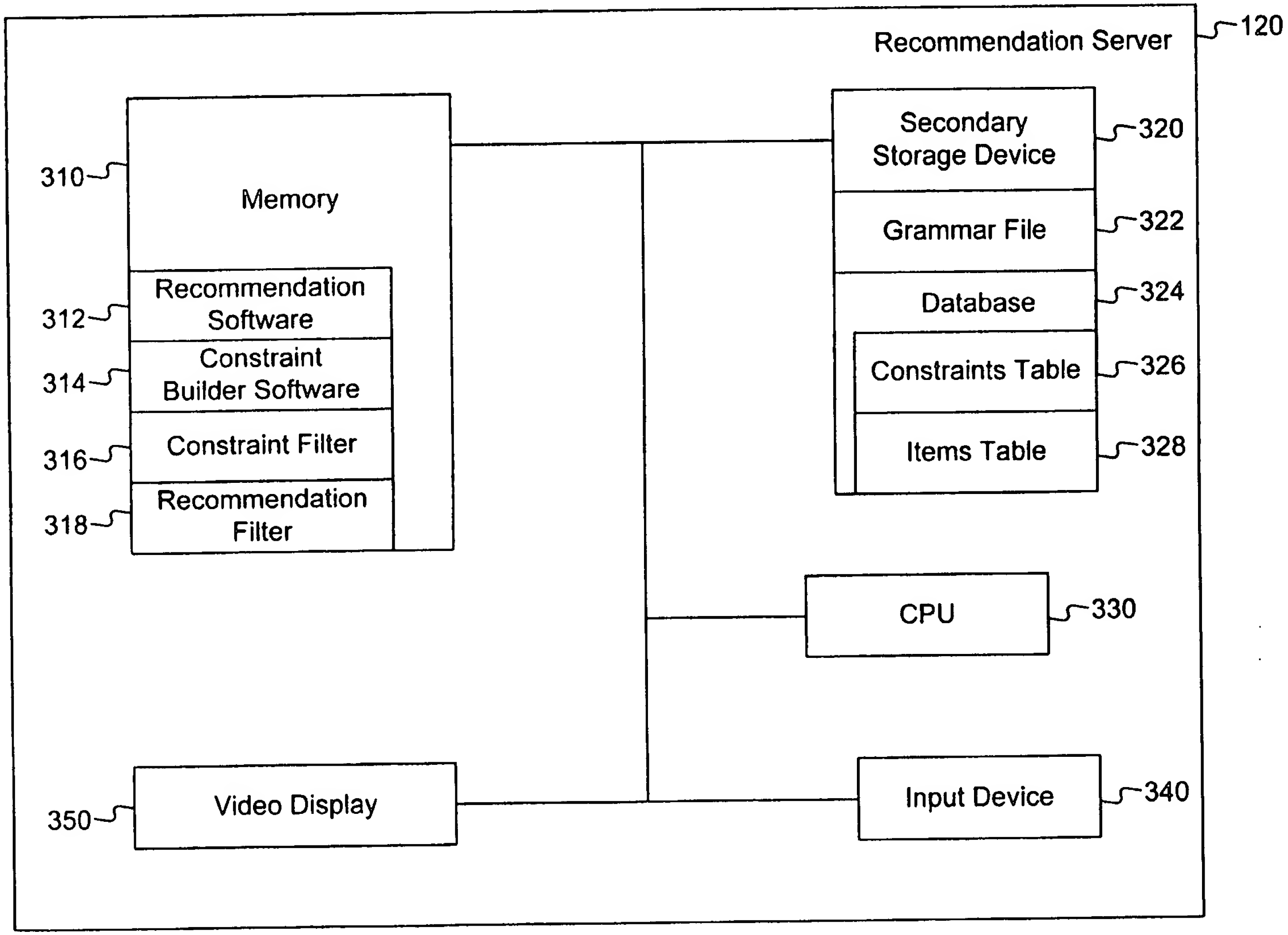

Recommendation Server
120
Memory
310
Recommendation Software
312
Constraint Builder Software
314
Constraint Filter
316
Recommendation Filter
318
Video Display
350
Secondary Storage Device
320
Grammar File
322
Database
324
Constraints Table
326
Items Table
328
CPU
330
Input Device
340

FIG. 3

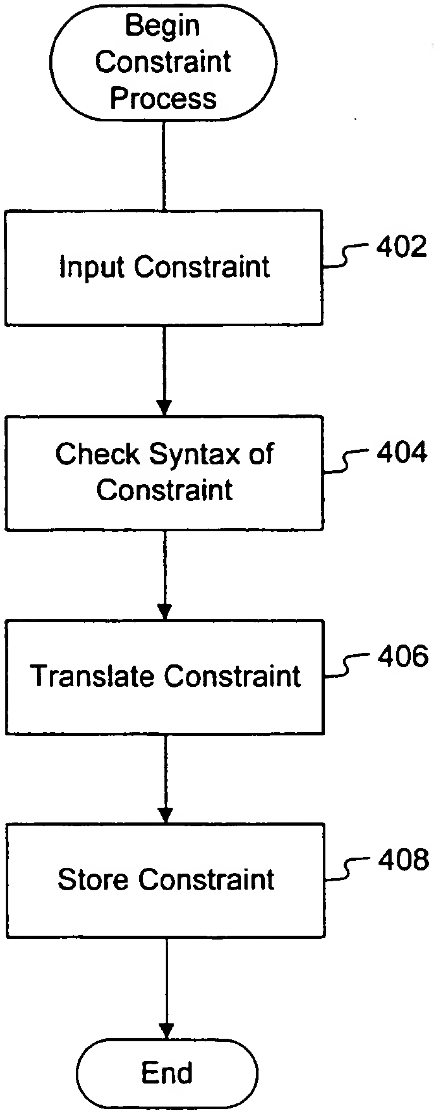

Begin Constraint Process
Input Constraint
402
Check Syntax of Constraint
404
Translate Constraint
406
Store Constraint
408
End

FIG. 4

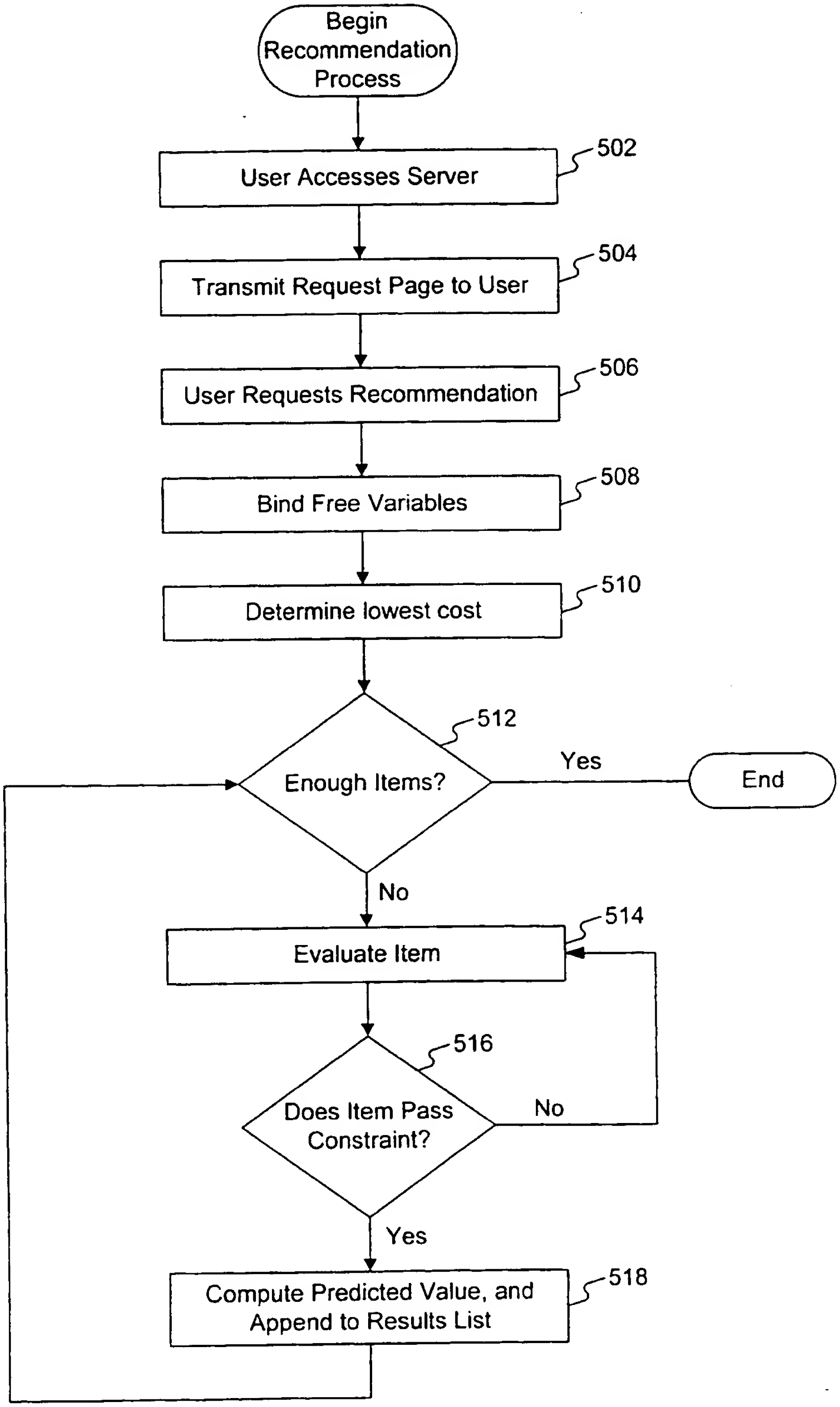

Begin Recommendation Process
User Accesses Server
502
Transmit Request Page to User
504
User Requests Recommendation
506
Bind Free Variables
508
Determine lowest cost
510
Enough Items?
512
Yes
End
No
Evaluate Item
514
Does Item Pass Constraint?
516
No
Yes
Compute Predicted Value, and Append to Results List
518

FIG. 5

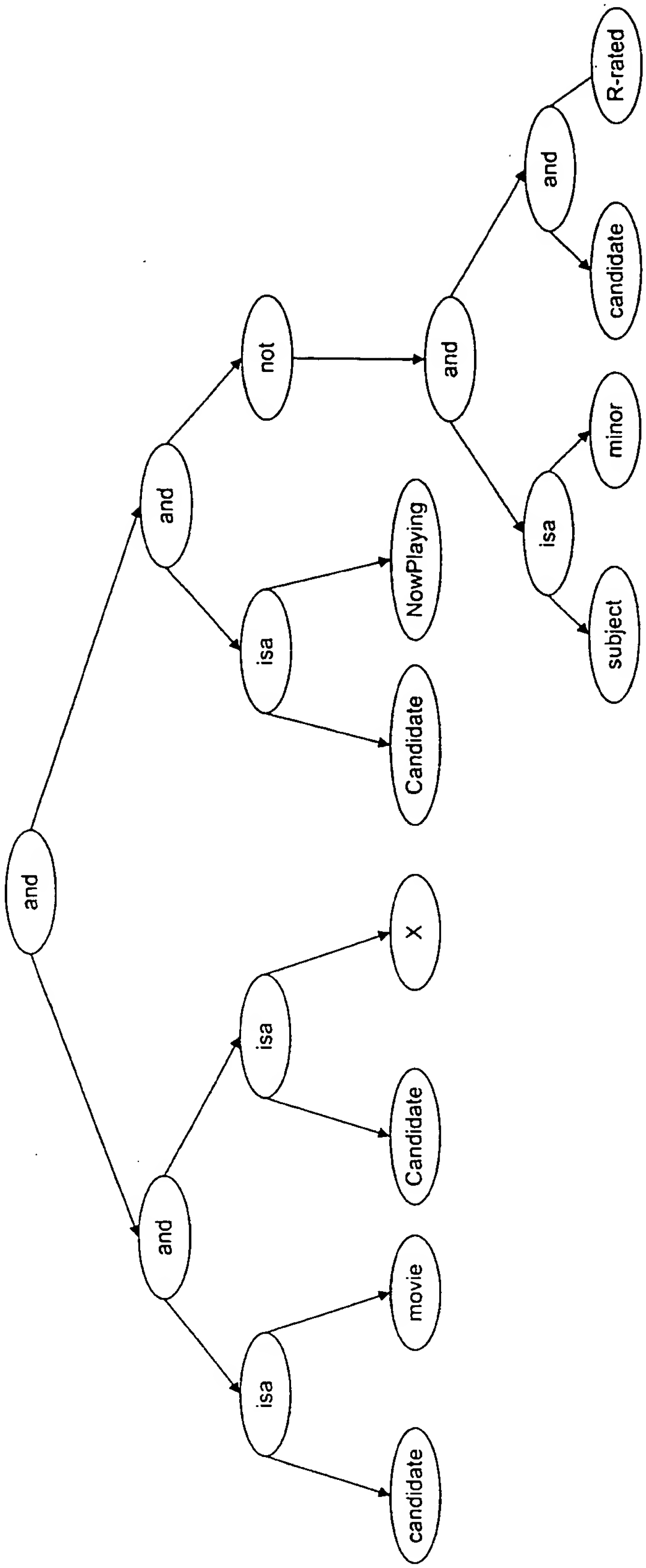
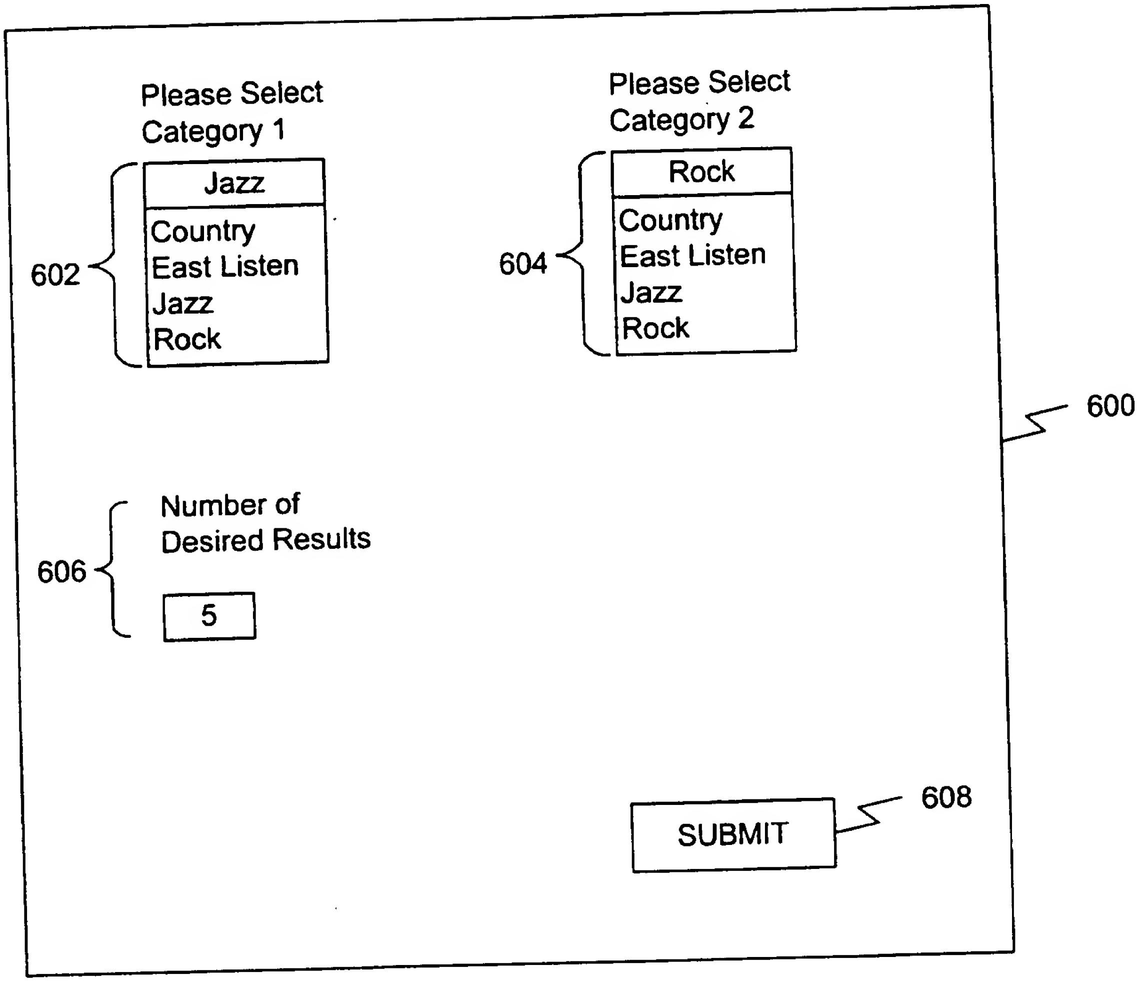

and
and
and
isa
isa
candidate
movie
Candidate
X
isa
Candidate
NowPlaying
not
and
isa
subject
minor
and
candidate
R-rated

FIG. 6A



Please Select Category 1

Jazz

Country
East Listen
Jazz
Rock

Please Select Category 2

Rock

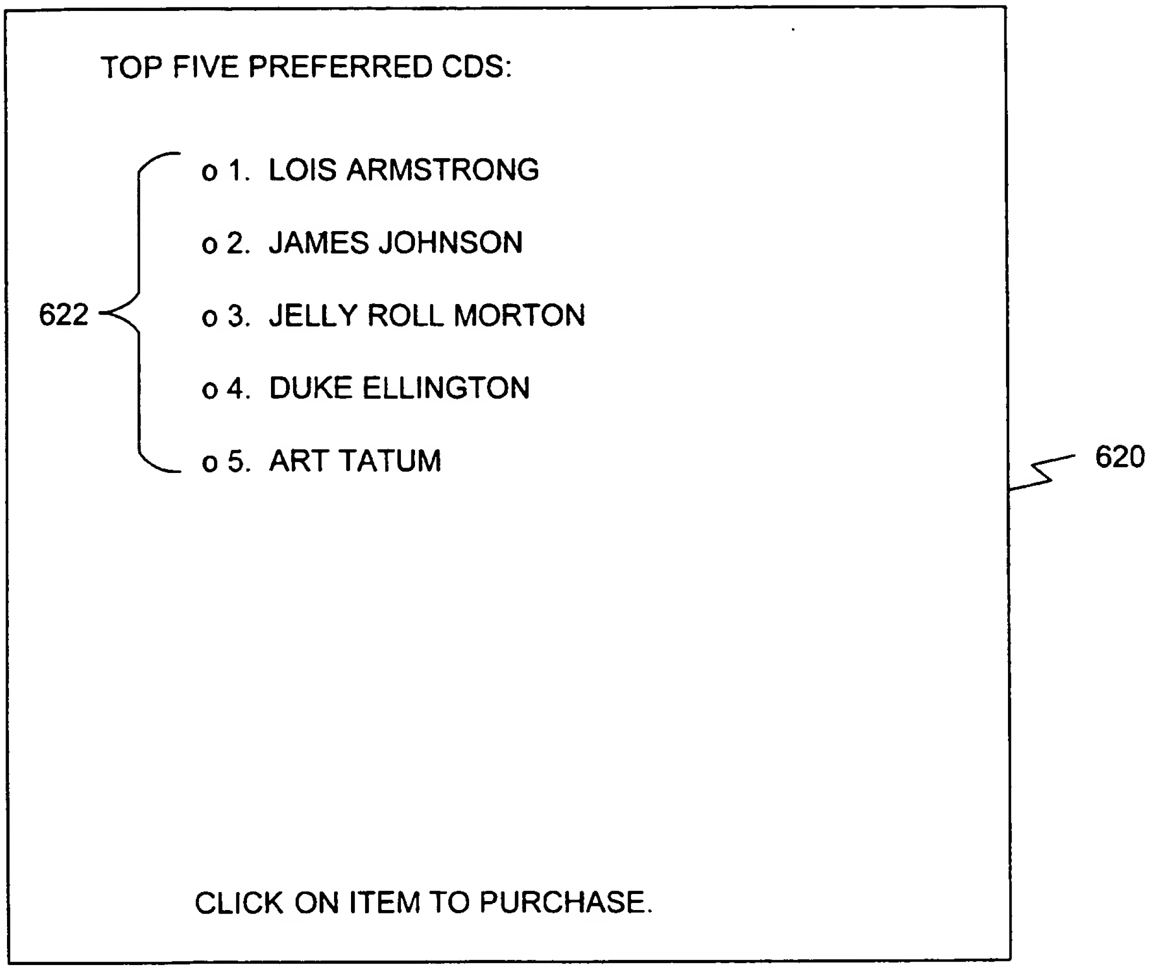
Country
East Listen
Jazz
Rock

Number of Desired Results

5

SUBMIT

FIG. 6B



TOP FIVE PREFERRED CDS:

o 1. LOIS ARMSTRONG

o 2. JAMES JOHNSON

o 3. JELLY ROLL MORTON

o 4. DUKE ELLINGTON

o 5. ART TATUM

CLICK ON ITEM TO PURCHASE.

FIG. 6C

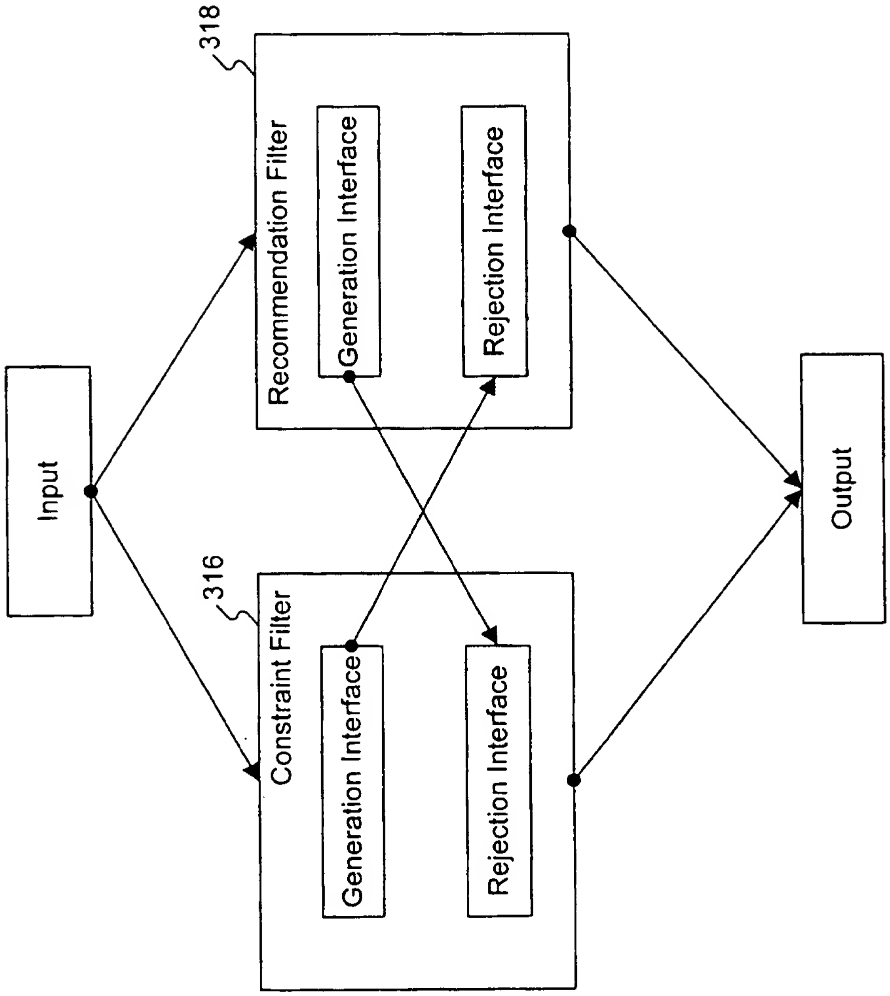

Input
318
Recommendation Filter
Generation Interface
Rejection Interface
316
Constraint Filter
Generation Interface
Rejection Interface
Output

FIG. 7